INTERNATIONAL DESIGN CONFERENCE - DESIGN 2000
Dubrovnik, May 23 - 26, 2000.

Problem-Adapted Mesh Generation With FEM-Features

Dipl.-Ing. Horst Werner, Prof. Dr.-Ing. Christian Weber,
cand. ing. Martin Schilke

Keywords
FEA, Feature-based Modelling, Object-Oriented Modelling Systems

Abstract
Today automatic meshing of CAD geometry is the most common method of FEM mesh generation. However, to get results of acceptable accuracy with universal meshing algorithms it is necessary to use rather small-sized elements which leads to high memory and CPU time consumption. Furthermore, the irregularity of automatically generatated meshes makes it difficult to create well-defined local areas with different material properties. A solution for this problem is the application of predefined building blocks ("FEM-Features") which contain algorithms for a problem-adapted mesh generation. These building blocks may be realised as "design objects" in an object-oriented modelling system. By employing interface objects, it is easy to combine design objects to form complex geometries. The technique is explained by an example.

1. Introduction

A crucial problem in Finite Element Analysis (FEA) is the mesh generation: on the one hand the mesh's quality is decisive for the accuracy of the results, on the other hand the generation of an efficient, problem-adapted mesh is usually very time-consuming. In the last few years the transfer of geometric data from CAD to FEM has become simple and computing power has become very cheap. Consequently the most common method nowadays is the application of a universal meshing algorithm on an existing CAD model. Such meshing algorithms are implemented in modern CAD systems (Pro/E, Solid Designer) as well as in FEM systems like MARC/Mentat or ANSYS. The drawbacks of this approach are due to the absence of case-specific “knowledge” in CAD systems and meshing algorithms. It can be observed that

- an automatically generated mesh needs considerably more elements to give reasonable results since a generic meshing algorithm can not take into account the peculiarities of a given geometry and therefore needs to use relatively small elements. Some meshing algorithms enlarge the elements towards the middle of large solids, but nevertheless these meshes can not compete with manually created ones in terms of element numbers.

- since most meshing algorithms use tetraeder elements, the interior structure of the mesh is more or less chaotic, which makes it impossible to define separate regions with different material properties. This means that certain cases must be treated as contact problems which need much more resources and can cause various complications.

- small structures in the CAD geometry which may be completely irrelevant for the calculation force automatic meshing algorithms to create very small elements and thus an explosive increase of the element number. In most cases it is necessary to generate a simplified CAD model for automatic meshing.

- today's CAD systems are not capable of handling non-geometric properties like mechanical interdependences between parts of a product. Although some systems allow the attributive definition of material properties, material properties and boundary conditions usually must be added manually to the FE model.

A different approach to automatic mesh generation is the assembly of the whole FEM model from predefined geometric elements ("Form Features"). For each of these Features a specific meshing algorithm creates a customized mesh. Going one step further, a kind of "Functional Features" can include information about material properties and boundary conditions so that a complete FEM model may be defined by the use of Features.

2. Existing Approaches

An early contribution on this subject was presented by [Unruh and Anderson 92]: They used an algorithm for automatic mesh generation based on a combination of Feature-based and BRep geometry representation. The advantage over conventional meshing algorithms was that it was able to cope with small cavities in the geometry. However, it was a conventional approach in so far as it used one generic algorithm to process complex geometry made up from different features. The algorithm used Delaunay triangulation and hence could only generate tetrahedron meshes.

[Schorcht et al. 99] have presented the "SPRING PROCESSOR" which offers – apart from libraries for common spring geometries and conditions of use – Features which can be combined in order to generate various special shapes of springs. The SPRING PROCESSOR is realized as extension of an existing FE system and is able to create meshes of eight-noded ("brick") elements. However, the technique used for the Feature-based mesh generation is not described in their paper.

It shall be mentioned that there also exist commercial tools based on case libraries for frequently occuring geometries and loadcases. The analysis can be performed automatically after filling in the parameters for geometry and load. However, these systems – as far as known to the authors – are not capable of combining geometry chunks to form arbitrary objects and thus can not be considered as Feature-based systems.

3. A Generalized Concept of FEM-Features

The approach presented in this paper makes use of specialized meshing algorithms for different Features. The crucial point in this case is the combination of the submeshes created by arbitrary FEM-Features. The connection is realised by means of two-dimensional "interfaces" which are subdivided by a rectangular or triangular lattice (for brick or tetrahedron elements respectively). The nodes of such an interface are shared by the two adjacent sub-meshes. The interfaces do not necessarily need to be planes: any two-dimensional surface can be used, provided that a reasonable numbering scheme for the nodes of the lattice can be found.

Generic Form-Features and Interfaces

The simplest way to realise FEM-Features is the implementation of Form-Features with algorithms for the generation of mere (mesh) geometry in a modeling system. These Form-Features must, in contrast to most existing approaches, have well defined interfaces. This requires the introduction of interfaces

as separate “Features” or “objects” into the modelling system. The Form-Feature’s meshing algorithm must take into account the subdivisions of all its interfaces.

An instance of a such a Feature contains all the elements and nodes belonging to its sub-mesh. The nodes belonging to the interfaces are stored separately in order to avoid redundant storage. Thus the interfaces do only contain nodes, but no elements. All features contain their own local coordinate systems and pointers to the adjacent interfaces (Form-Features, respectively, in the case of interfaces).

When a Feature is created or updated, it aligns all adjacent interfaces according to its own position and orientation. The interfaces subsequently align other affected Features so that the structure of the whole assembly is kept consistent. A similar mechanism can be implemented to assure that a Feature's mesh is updated when the subdivision of an adjacent interface changes.

Modifying Features

Modifying Features like holes or notches can not be treated in the way described above since they do not add geometry. There seem to be two possible solutions to this problem:

- Modifying Features with universal meshing algorithms: These Features modify an existing mesh independently of its origin. They need algorithms which identify all affected elements and nodes and realign or remove them in order to obtain the desired geometry. The drawback is that such algorithms must be very sophisticated in order to assure that they work for all kinds of meshes.

- Modifying Features with specialized meshing algorithms: These use different algorithms depending on the type of Form Feature they are applied to. They might even completely replace the Form Feature’s meshing algorithm, which is equivalent to having a separate “modified Form Feature”. The advantage is that the algorithm is simple and can easily create an optimized mesh. The drawback is, of course, that a separate algorithm is needed for almost each combination of Form Feature and modifying Feature.

Advanced FEM-Features

Apart from the quick mesh generation, Feature-based modelling for FEA allows to include material properties and distribution, which must be added manually in conventional systems. This is especially important if the material is anisotropic or a compound of various materials is used. Furthermore, several analysis options can be automatically selected based on knowledge which is stored with specific Features (e.g. the applicable equivalent stress hypothesis for a material).

4. Integration into an Object-Oriented Modelling System

The implementation of Features by means of Object-Oriented Programming has become quite popular in the last few years due to numerous advantages. The extension of the Feature idea towards a “design object” allows the integration of functional information and behaviour description in a completely explicit product model [Weber and Werner 99].

This concept has been realized in an Object-Oriented modelling system (“Ligo”) developed at the Institute of Enginering Design in Saarbrücken. A characteristic of the Ligo system is the use of “interface objects” for information exchange between design objects. Thus the realization of FEM-Features and their interfaces in Ligo is very easy. Furthermore, the concept of “design objects” allows to determine the boundary conditions for parts of a complex assembly by means of analytical calculations and to create a complete FEM input file for this part, including geometry, material properties and boundary conditions.

5. Example

Fig. 1 shows a mesh for a motorcycle's rear arm generated by a conventional meshing algorithm. The rear arm is designed to be made of fibreglass compound with a foam core in order to integrate the functions of a guidance and a spring and therefore has an additional support in front of the rear arm pivot. Since fibreglass is an orthotropic material, its orientation must be defined in the FE model. For parts with large curvatures the material orientation will be element dependent. This is almost impossible to realize with tetrahedron elements as seen in Fig. 1. Also it is not possible to model the foam core since due to the "chaotic" mesh there is no well-defined geometry inside the solid.

The mesh consists of 4285 elements but still makes a rather crude impression: The holes and the arc in the center are clearly polygonal.

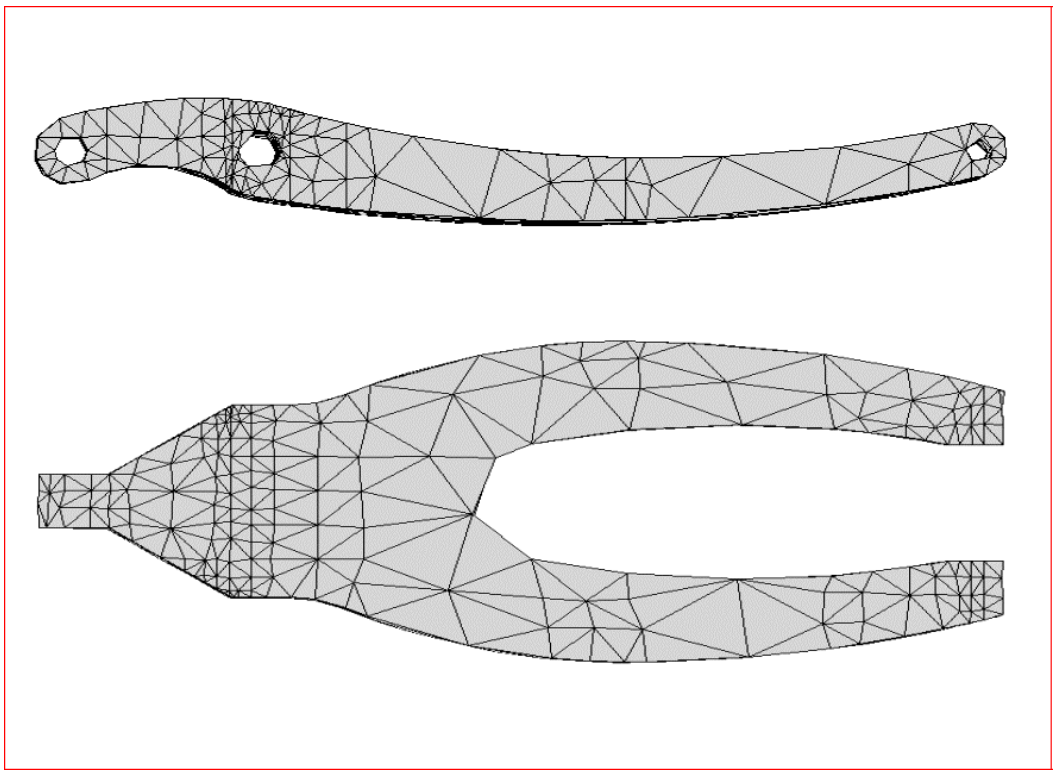

Fig. 1: Motorcycle rear arm, conventional meshing algorithm

Fig. 2 shows a similar geometry generated by FEM-Features in a stand-alone object-oriented modeller[1]. The whole model is made up from three types of Form Features: "beam" (curved in two planes with rectangular cross section), "hole" and "wishbone". These Features can be connected to each other by the "rectangle" interface. The Feature "hole" also contains a "cylinder" interface for the connection to a bush or a bolt. The employed Feature composition is shown in Fig. 3.

The resulting mesh consists of only 688 elements and still offers a finer representation of the holes and arcs than the mesh in Fig. 1. The material orientation can be automatically written into the model file and the definition of a foam core in the interior is easy to accomplish due to the simple mesh structure.

[1] The design objects containing the respective meshing algorithms were implemented in Visual Basic based on Ligo routines due to performance reasons. Although an implementation in Ligo's own programming language would have been easier, the additional overhead of Ligo interpreting the meshing algorithms would have made the handling of large element numbers too awkward.

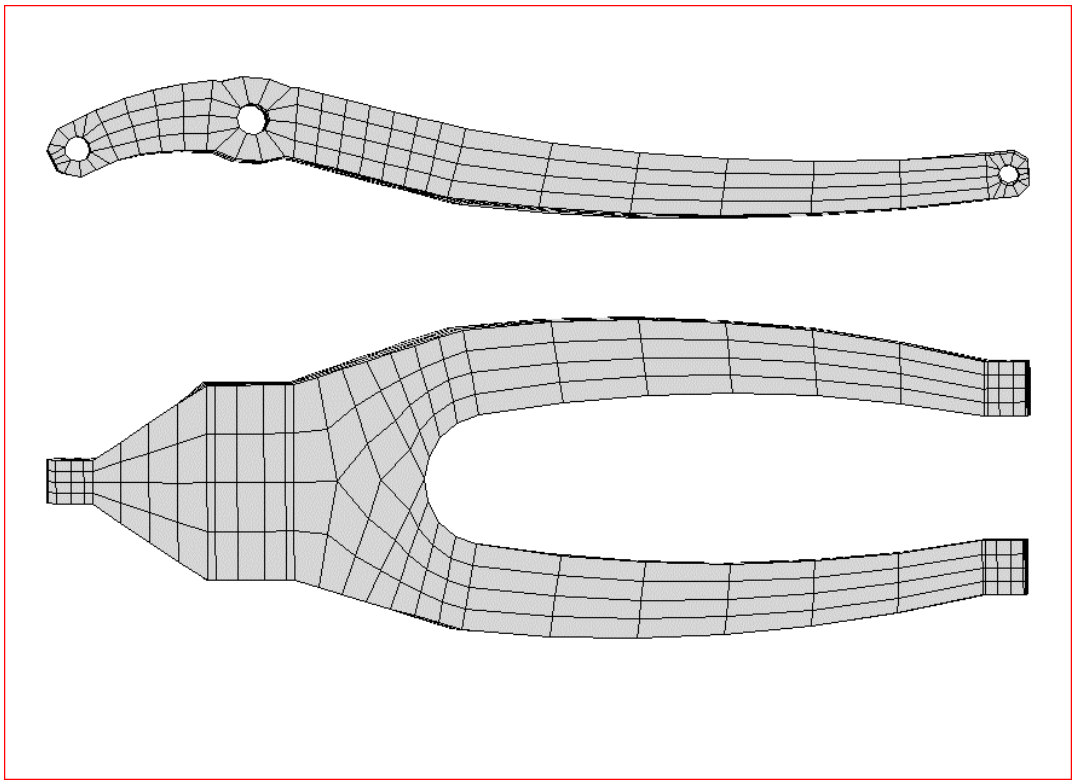

Fig. 2: Motorcycle rear arm, mesh generated by means of FEM-Features

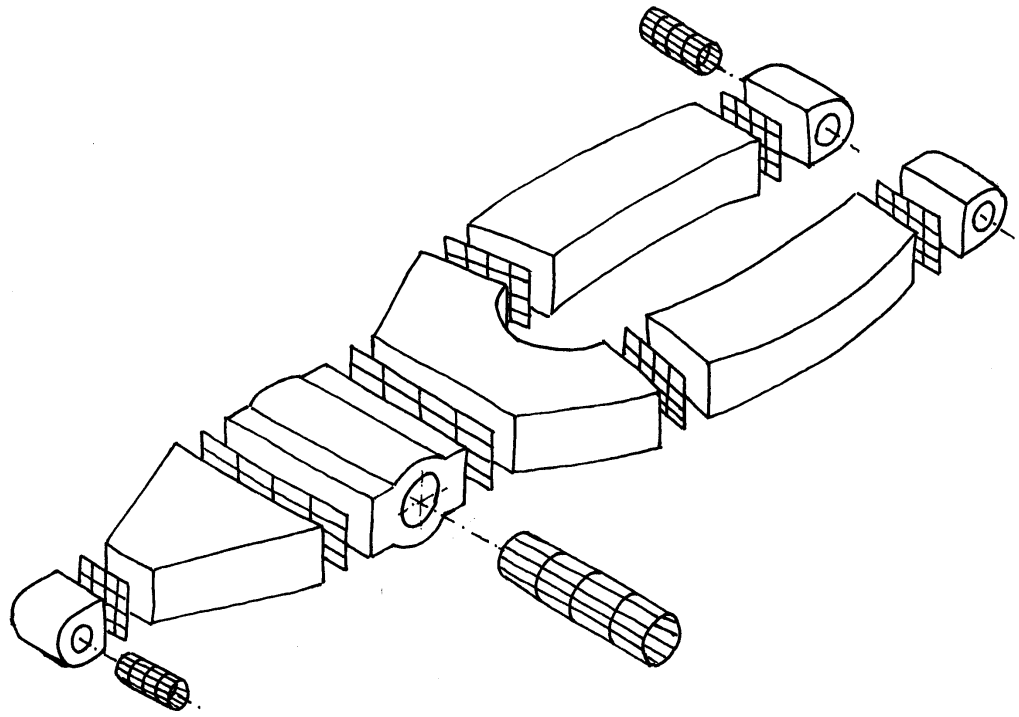

Fig. 3: Feature composition for the rear arm

A further advantage of having submeshes with well-defined geometry is that adjacent parts can be included into the mesh (if no relative displacement is expected at the surfaces) in order to avoid the problems which usually come along with contact analysis. Fig. 4 shows a “hole” Feature of a fibre-glass structure connected to a rubber bush which is connected with a steel bolt. All three parts form

one mesh. The axis of the bolt can be fixed so that on the one hand rotation is still possible, but on the other hand artefact stress peaks are avoided in the fibreglass part.

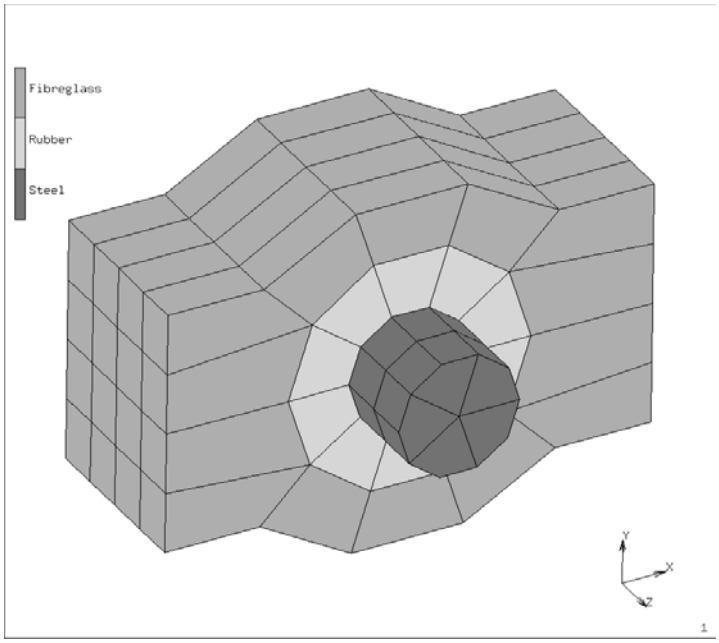

Fig. 4: Steel bolt and rubber bush integrated into mesh

Conclusion

A new approach is shown to the automatic generation of meshes for FEA, which has a number of advantages over conventional (universal) meshing algorithms. The mesh is composed of submeshes generated by specialized algorithms for the respective Form Features. Interfaces assure that the sub-meshes fit together, therefore these interfaces must be represented in the model as separate Features/objects. The approach is especially suitable for object-oriented modelling systems since the meshing algorithms can be directly assigned to the respective Form Features/objects.

References

V. Unruh, D. C. Anderson, "Feature-Based Modeling for Automatic Mesh Generation", Engineering with Computers 8, Springer-Verlag, New York, 1992

H.-J. Schorcht, U. Kletzin, D. Micke, "Spring Processor – An Example of an Object-Oriented FEM Application", Proceedings of ICED 99, Vol. 1, pp. 445-448, TU München, 1999

C. Weber, H. Werner, "Implizite und explizite Bestandteile des Produktmodells und ihre Bedeutung für die Entwicklung von CAx-Systemen", Symposium Fertigungsgerechtes Konstruieren, Schnaittach, 1999

Address

Dipl.-Ing. Horst Werner, Prof. Dr.-Ing. Christian Weber, cand. ing. Martin Schilke
Chair of Enginering Design/CAD, Saarland University
PO Box 15 11 50, D-66041 Saarbrücken, Germany
Phone: (++49) 681 / 302 - 3387, - 3075
Fax: (++49) 681 / 302 - 4858
E-Mail: werner@cad.uni-sb.de, weber@cad.uni-sb.de, schilke@cad.uni-sb.de